UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,012	11/21/2003	Eero Kaappa	915-006.029	2782

4955 7590 11/22/2006

WARE FRESSOLA VAN DER SLUYS &
ADOLPHSON, LLP
BRADFORD GREEN, BUILDING 5
755 MAIN STREET, P O BOX 224
MONROE, CT 06468

EXAMINER	
ROSE, HELENE ROBERTA	
ART UNIT	**PAPER NUMBER**
2163	

DATE MAILED: 11/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/720,012	**Applicant(s)** KAAPPA, EERO	
	Examiner Helene Rose	**Art Unit** 2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *August 31, 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-10,13,14 and 16-22* is/are pending in the application.

4a) Of the above claim(s) *1,11,12 and 15* is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-10,13,14 and 16-22* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 November 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)□ Some * c)□ None of:

1.☒ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

Detailed Action

1. In response to communications, Claims 2-10, 13-14 and 16-17 is presently pending; Claims 1, 11-12, and 15 were cancelled; Claims 18-22 is added.

2. Applicant's arguments, filed on 8/31/2006, with respect to claims 2-10, 13-14, and 16-22 have been considered, but are not persuasive.

Claim Objections

3. In view of claims 12, 15, and 16, being objected to because claims included the limitation "and/or". **Examiner withdraws the pending rejection based on applicant's amendment to claim 16, to recited "and" and cancellation of claims 12 and 15**.

4. In view of the objected claims, Claims 1, 2, 3, 11, 12 and 15 being objected to because of the following having improper citations pertaining to commas, colon's, and semicolons after claim limitations. **Examiner withdraws the pending rejection based on applicant's amendments to claims to claims 2 and 3 and cancellation of claims 1, 11, 12, and 15**.

However, Claims 16,19 and 20 are objected to because of the following informalities: Claims 16,19, and 20 have improper citations pertaining to colon's, and semicolons after claim limitations. Examiner is unclear on whether or not it should contain a comma or a semi-colon, because the following citations indicated are not consistent with one another, wherein some limitations cite a comma and some limitations cite a semi-colon. Appropriate correction is required.

Claim Rejections – 35 U.S.C – 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said

> subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2-10, 13-14, 16-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Andrews et al (US Patent No, 6,105,062, Date of Patent: August 15, 2000) in view of Jőnsson et al (Non-Patent Literature "SyncML – Getting the Mobile Internet in Sync", Ericsson Review No. 3, 2001).

Claim 2:

Regarding claim 2, Andrews teaches checking whether said parent object has already one or more associated objects being directly arranged subordinate to said parent object (column 3, lines 26-37, wherein schema associated with the destination directory tree is compared to the schema associated with the source directory tree, if the schemas differ, classes that do not exist in the destination directory schema lacks attributes present in the source directory schema, such attributes are added to the destination directory schema so that all data associated with objects in the subtree can be moved from the source directory to the destination directory, wherein comparing is equivalent/method of checking, Andrews);

wherein if at least one of said one or more already existing objects is of said run-time object type (column 3, lines 39-42, wherein it is determined whether the move will cause naming conflicts between objects in the sub-tree and existing objects in the destination directory tree, Andrews), rejecting said defining of said **new** object (column 8, lines 19-26, wherein analyze the name of the root object of the subtree and determines whether the name of object will conflict with a name of object will conflict with a name of any object in destination directory tree, wherein if a naming conflict exists the object name of object is modified to eliminate the conflict, wherein eliminate is equivalent to rejecting, Andrews).

Claim 3:

Regarding claim 3, Andrews teaches checking whether said parent object is of said fixed object type (column 8, lines 29-33, wherein type object would violate containment rules associated with destination directory, the object type of object is changed to an UNRESTRICTED object, type, wherein UNRESTRICTED object type is a special object type that can exist anywhere in the tree without violating containment rules, wherein the UNRESTRICTED object type is interpreted to be fixed because it doesn't violate rules, wherein the containment rules prohibit certain parent/child relationships among the objects, wherein rules are predefined);

checking whether said parent object has already one or more associated objects being directly arranged subordinate thereto (column 3, lines 14-17, wherein identification of the destination location includes specification of a destination object to which the parent object of the subtree will be immediately subordinate after the move, Andrews); and

checking whether said at least one object is of said fixed object type (column 8, lines 29-33, wherein type object would violate containment rules associated with destination directory, the object type of object is changed to an UNRESTRICTED object, type, wherein UNRESTRICTED object type is a special object type that can exist anywhere in the tree without violating containment rules, wherein the UNRESTRICTED object type is interpreted to be fixed because it doesn't violate rules, wherein the containment rules prohibit certain parent/child relationships among the objects, wherein rules are predefined, Andrews);

wherein if said parent object has no already associated objects (column 7, lines 32-41, wherein the schema contains containment rules which govern the permissible **parent/child** relationships among different types of objects, wherein one problem with moving subtree from source directory tree may include attributes which are not defined in the corresponding classes of the schema associated with the destination directory tree, wherein the schema is interpreted to be a pattern that represents the data's model defining the elements, i.e. or objects, their attributes, i.e. or properties, and the relationships between the different elements, in which the schema is interpreted to be fixed and said parent object, Andrews) and said **new** object are of said fixed objects type (column 7, lines 42-48, wherein to ensure that all objects are moved, the system/method determines that attributes exist in classes in sour directory tree that do not exist in corresponding classes associated with destination directory tree such attributes will be addresses to the class definitions of the schema associated with the destination directory tree, Andrews),

concentrating said parent object and said **new** object by replacing (column 9, lines 41-47, wherein code is utilized for renaming of an object, which is interpreted to be replacing, Andrews) said parent object and said **new** object with one combined new object being of a fixed object type (column 8, lines 48-62, wherein subtree can be restored from the intermediate location to a location that is subordinate to the root-most object of the tree destination tree, i.e. object, wherein subtree is moved to its final target location represented by a subtree, object is then modified from unrestricted object type to an object type that will conform to the containment rules of destination directory such as an organization unit type object, wherein a subtree of objects can be seamlessly and automatically moved from a source location to a destination location eliminating the need to manually delete and recreate objects, wherein recreate is equivalent to new, Andrews).

Claim 4:

Regarding claim 4, Andrews teaches coding at least a part of a description document being based on said definition of said **new** object (column 12, lines 31-32, Andrews) and comprising information relating to said **new** object and said properties of said **new** object (column 12, lines 47-55, wherein code is utilized for a prune operation, wherein prune is equivalent to part, wherein source tree and destination trees are defined, Andrews);

wherein said coded description document allowing for generating a hierarchical node structure for storing said management related information (column 12, lines 4-32, wherein code to move subtree to this intermediate location, Andrews).

Claim 5:

Regarding claim 5, Andrews discloses all the limitations above. However, Andrews does not disclose wherein said hierarchical object structure is an information being part of the device description framework and said hierarchical node structure is a management tree employed for device management according to the synchronization markup language device management standard defined by the SyncML Initiative.

On the other hand, Jőnsson discloses wherein said hierarchical object structure is an information being part of the device description framework and said hierarchical node structure is a management tree employed for device management according to the synchronization markup language device management standard defined by the SyncML Initiative (**SEE** non-patent literature titled – "SyncML

Getting the mobile Internet in sync", **PAGE 113,** wherein subtitle "SyncML Framework", wherein SyncML defines the format for synchronization data as well as conceptual data synchronization framework and data synchronization protocol, wherein the scope of the SyncML framework consist of the Sync ML format and conceptual SyncML adapter and SyncML interface, wherein the framework is useful for describing the particular system model associated with the SyncML implementations, Jőnsson). It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate Jőnsson teachings into Andrews's system to synchronize how data is organized in respective data stores maintained by the devices.

A skilled artisan would have been motivated to do so to include SyncML's; data modeling and device management tree to provide end users the freedom and greater cost efficiency of being able to synchronize data across a broader range of devices and lowering the service threshold for the customers.

Claim 6:

Regarding clam 6, the combination of Andrews in view of Jőnsson teaches wherein said description document is at least a part of a device description framework document, said device description framework document being an extended markup language encoded document being encoded in accordance with a corresponding description framework document type description (**SEE** non-patent literature titled – "SyncML Getting the mobile Internet in sync", **PAGE 111,** wherein subtitle "Open industry-standard technology", wherein syncML uses the industry standard extensible markup languages for specifying the messages that synchronize devices and applications using either plan text or the wireless binary XML, binary encoding technique employed by WAP; and **PAGE 14,** wherein subtitle "Optimized for the mobile environment", wherein exchanged data is generally binary-encoded and so forth; and SEE section heading " Representation specification", wherein the representation specification specifies and XML document type description, i.e. DTD, which allows the representation of all information required to perform synchronization including data, metadata, and commands, Jőnsson).

Claim 7:

Regarding claim 7, the combination of Andrews in view of Jőnsson teaches a software tool for defining a hierarchical structure consisting of a plurality of entities comprising program portions for carrying out the operations when said program is implemented in a computer program for being executed on a processing device, a networked device, a networked server, a terminal device or a communication terminal device (column 5, lines 45-52, wherein directories are frequently distributed, meaning portions, of the directory are stored on different servers, such portions are frequently referred to as partitions, server may have one or more partitions while other servers on the network contain different partitions of the network, Andrews).

Claim 8:

Regarding claim 8, the combination of Andrews in view of Jőnsson teaches a computer program product for defining a hierarchical structure consisting of a plurality of entities, comprising loadable program code sections for carrying out the operations when said computer program is executed on a processing device, a networked device, a networked server, a terminal device or a communication terminal device (column 5, lines 54-64, wherein it operates in a conventional computer such as those available from any number of commercial computer manufacturers in a variety of different programming languages, wherein loadable is equivalent to implemented, column 9, lines 41-47;column 10, lines 26-31; column 10, lines 48-64; column 12, lines 4-9, and lines 36-44, wherein lines of pseudo code is shown and demonstrated to execute different sections, such as renaming an object, comparing schemas and adding, prune operations, Andrews).

Claim 9:

Regarding claim 9, the combination of Andrews in view of Jőnsson teaches wherein said computer program product comprises program code sections stored on a computer readable medium for carrying out the method of claim 1 when said computer program product is executed on a processing device, a networked device, a networked server, a terminal device or a communication terminal device (column 3, lines 61-65, wherein a system includes a processor that is operative to execute program instructions and so forth, Andrews).

Claim 10:

Regarding claim 10, the combination of Andrews in view of Jőnsson teaches a computer data signal embodied in a carrier wave and representing a program which, when executed by a processor

(**SEE** non-patent literature titled – "SyncML- Getting the mobile Internet in sync", **PAGE 110**, wherein the SyncML initiative is a universal standard synchronization across networks, devices, and applications from any vendor, the initiative is supported by Ericsson, IBM Lotus, Nokia, Openwave, wherein Openwave is equivalent to carrier wave; also **PAGE 112,** wherein subtitle "Optimized for the mobile environment, wherein syncML has been designed to be robust enough to cater for the relatively high network latency in mobile networks which increase as packet data techniques such as general packet radio service, i.e. GPRS, Jőnsson).

Claim 13:

Regarding claim13, the combination of Andrews in view of Jőnsson teaches wherein each object of said plurality of objects is allowed to have only one directly subordinate arranged object which has said run-time object type (column 1, lines 30-37, wherein one directory service product, Novel Directory categorizes each object in the tree as either a container object or leaf object, wherein a container object can be a parent to other container objects and to zero or provide a logical organization to the tree, while the leaf objects represent actual network entities such as servers, printers, facsimile machines, and users; column 4, lines 41-65, wherein the system and is more defined, Andrews).

Claims 14 and 17:

Regarding claims 14 and 17, the combination of Andrews in view of Jőnsson teaches wherein said hierarchical object structure has a concentrated object which has said fixed object type (Referring to column 8, lines 29-33, wherein type object would violate containment rules associated with destination directory, the object type of object is changed to an UNRESTRICTED object, type, wherein UNRESTRICTED object type is a special object type that can exist anywhere in the tree without violating containment rules, wherein the UNRESTRICTED object type is interpreted to be fixed because it doesn't violate rules, wherein the containment rules prohibit certain parent/child relationships among the objects, wherein rules are predefined, Andrews), wherein said concentrated object is constructed from a parent object and a child object (column 3, lines 14-17, wherein the apparent object subtree will be immediately subordinate after move, Andrews), wherein said parent object is an object which is directly arranged subordinate to said child object (**REFER** to claim 12, wherein this limitation has been addressed, Andrews), **and**

wherein said concentrated object is implemented **if**:

said parent object has said fixed object type (column 5, lines 1-9, wherein containment rules prohibit certain types of parent/child relationships among different types of objects, which is interpreted to be fixed, Andrews);

said child object has said fixed object type (column 5, lines 1-9, wherein containment rules prohibit certain types of parent/child relationships among different types of objects, Andrews); and

said child object is the only object, which is directly arranged subordinate to said parent object (column 6, lines 49-58, wherein subordinate to object, diagram 42, are the organization type container objects, diagrams 44 and 43, object, i.e. diagram 42, has a parent relationship to its children objects, which are diagrams 43 and 44, Andrews).

<u>**Claim 16**</u>:

Regarding claim 16, the combination of Andrews in view of Jőnsson teaches wherein two or more child nodes which have said run-time node type and which have a same parent node have a common format (Figure 2, wherein diagram 44 is subordinate to diagrams 45 and 46, wherein diagram 44 is superior and wherein diagrams 43 and 44 represent the two child nodes which have the same run-time node, as defined in column 6, lines 54-56, to be wherein it may represent a different country in which the organization maintains an office, but have the same parent node, diagram 42, Andrews),

wherein said parent node is directly subordinate arranged to said **two** or more child nodes (Figure 2, wherein diagram 44 is subordinate to diagrams 45 and 46, wherein diagram 44 is superior and wherein diagrams 43 and 44 represent the two child nodes which have the same run-time node, as defined in column 6, lines 54-56, to be wherein it may represent a different country in which the organization maintains an office, but have the same parent node, diagram 42, Andrews), **and**

wherein said common format determines that management related information distributed among said two or more child nodes having said run-time node type relates to the same function of **at least one out of** said managed mobile communication enabled device and/or application running on said managed mobile communication enabled device (**SEE** non-patent literature titled – "SyncML- Getting the mobile Internet in sync", **Page 115** – wherein the SyncML commands do not fully define the semantics of the SyncML operation, wherein for example adding a document to an email system database might have very different semantics from that of adding a transaction request to a queue, wherein the type of data being synchronized determines the semantics of the SyncML operation, in which this means that it is for the an originator to request an operation of a particular recipient, in which the recipient returns an error response status code, wherein the exact equivalence of data items is defined by the data synchronization model an can be limited by the capabilities of the device that hold the two data collections and so forth, Jőnsson).

Claim 18 **(New)**:

Regarding Claim 18, the combination of Andrews in view of Jőnsson teaches wherein a method, comprising:

defining at least one new object to be associated to a parent object arranged directly superordinate to said new object, and part of a hierarchical object structure comprising a plurality of objects being hierarchically associated (column 2, lines 16, wherein due to a company reorganization it may be useful to move a subtree from one location within the tree to another location within the tree to accurately reflect a new reporting hierarchy; column 3, lines 14-17, wherein identification of the destination location includes the specification of a destination object to which the parent object of the subtree will be immediately subordinate after the move, the object type of one or more of the objects in the subtree is changed such that the location of the objects in the destination directory tree and grafted at their specified location in the destination directory tree, in which the changes, grafted, and moving is equivalent to arranged; column 9, lines 45, wherein newname = oldname +_+ tree name, Andrews);

wherein said plurality of objects comprises different types of objects out of a group including at least a fixed object (Referring to column 8, lines 29-33, wherein type object would violate containment rules associated with destination directory, the object type of object is changed to an UNRESTRICTED object, type, wherein UNRESTRICTED object type is a special object type that can exist anywhere in the tree without violating containment rules, wherein the UNRESTRICTED object type is interpreted to be fixed because it doesn't violate rules, wherein the containment rules prohibit certain parent/child relationships among the objects, wherein rules are predefined, Andrews), a run-time object (column 5, lines 54-64, wherein running programs executing in a run-time environment, Andrews), a leaf object (column 4, lines 46-48, wherein objects represent actual entities are referred to as "leaf" objects), and a link object (columns 11-12, lines 1-4, wherein ensuring attributes links to other data in the sub-tree are not lost, and such attribute links can be modified, Andrews);

wherein said fixed object type has a fixed title, and wherein said run-time object has a title defined during run-time (column 4, lines 54-55, wherein organization type unit container entitled Engineering, wherein entitled is interpreted to be title, wherein entitled is defined to be "to given a title to", Andrews);

checking said type of said parent object (column 3, lines 26-37, wherein schema associated with the destination directory tree is compared to the schema associated with the source directory tree, if the schemas differ, classes that do not exist in the destination directory schema lacks attributes present in the source directory schema, such attributes are added to the destination directory schema so that all data associated with objects in the subtree can be moved from the source directory to the destination directory, wherein comparing is equivalent/method of checking, Andrews); and

wherein if said parent object is of said fixed object type, defining said new object having a type out of a group comprising said fixed object(column 6, lines 38-47, wherein removing large numbers of objects is equivalent to fixed objects), said run-time object said leaf object (column 5, lines 54-64, wherein running programs executing in a run-time environment and column 4, lines 46-48, wherein objects represent actual entities are referred to as "leaf" objects, Andrews) and said link object (columns 11-12, lines 1-4, wherein ensuring attributes links to other data in the sub-tree are not lost, and such attribute links can be modified, Andrews); and

wherein if said parent object is of said run-time object type, defining said new object having a type out of a group comprising said fixed object, said leaf object and said link object (column 4, lines 45-65, wherein leaf object and linking of groups is defined; column 6, lines 62-65, wherein branch is equivalent to linking and wherein one or more leaf objects, Andrews);

defining properties of said new object (column 9, line 45, wherein newname = oldname +_+ tree name, Andrews);

wherein said plurality of objects serves as template object for deriving corresponding nodes ((Figures 4A and 4B, all features; column 12, lines 45-55, wherein a result is produces by if a source tree and destination tree are the same tree and the move is within the same schema but the move would otherwise prohibit because of containment rule violation, wherein certain steps can be omitted, if the move of the subtree will cross schemas but would not be prohibited because of containment rule violations it is possible certain steps can be omitted, in which the following is interpreted to be deriving, and wherein deriving is equivalent to producing a result, Andrews);.

wherein said hierarchical object structure serves as a template structure for deriving hierarchical node structures (column 6, lines 53-67, wherein organization types objects, diagrams 43 and 44 may represent a logical or physical division of an organization, diagrams 43 and 44 may each represent a different country in which organization maintains an office, subordinate to object, diagram 44 are organization unit type container objects, diagrams 45 and 46, wherein additional objects would typically be subordinate to Organization Unit-Type objects, diagrams 45 and 46, wherein each branch would typically terminate with one or more leaf objects that represent network entities); and

wherein said hierarchical node structure comprising a plurality of said nodes is employed for distributing management related information of an electronic device among said plurality of nodes, certain parts of said management related information being assigned to at least one of said plurality of nodes (column 4, lines 37-44, wherein hierarchical tree structure simplifies administration entities, wherein a directory service products that uses such a tree like structure for network entity administration is Novell Corporation; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network, wherein frequently distributed meaning portions are stored on different servers; column 10, lines 11, wherein hard disk is equivalent to an electronic device, Andrews).

Claim 19 (**New**):

Regarding Claim 19, the combination of Andrews in view of Jőnsson teaches an apparatus, comprising:

a processing unit for defining at least one new object to be included into a hierarchical object structure comprising a plurality of objects being hierarchically associated (column 2, lines 36-40; column

5, lines 14-19, wherein a schema may define the classes for a whole directory, or a portion of a directory, wherein the schema includes the classes as well as the containment rules defining the parent/child relationships that are permitted along objects, Andrews), said hierarchical object structure being constituted by a plurality of objects being hierarchically associated (Figures 2 and 3, all features, wherein a tree-like hierarchically structures are defined; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network Andrews);

a memory unit (columns 3-4, lines 61-67 and lines 1-4, wherein the system includes a processor that is operative to execute program instructions, and so forth, wherein memory is a computer physical workspace that stores the instruction programs and data needed to accomplish the task executed by the processor, Andrews); and

a communication interface (columns 3-4, lines 61-67 and lines 1-4, wherein the system includes a processor that is operative to execute program instructions, and so forth, wherein memory is a computer physical workspace that stores the instruction programs and data needed to accomplish the task executed by the processor, Andrews);

wherein said plurality of objects comprising different types of objects out of a group (column 4, lines 51-65, wherein example defines a particular group of an organization such as engineering group may be organized in a Organization type Unit container objected entitled Engineering and so forth, Andrews) including at least a fixed object (column 6, lines 38-47, wherein removing large numbers of objects is equivalent to fixed objects, Andrews), a run-time object (column 5, lines 54-64, wherein running programs executing in a run-time environment, Andrews) a leaf object (column 4, lines 46-48, wherein objects represent actual entities are referred to as "leaf" objects, Andrews) and a link object (columns 11-12, lines 1-4, wherein ensuring attributes links to other data in the sub-tree are not lost, and such attribute links can be modified, Andrews);

wherein said fixed object type has a fixed title, and wherein of said run-time object has a title defined during run-time (**<u>Refer</u>** to claim 18, wherein this limitation as already been addressed);

wherein said processing unit is configured for;

checking said type of said parent object (**Refer** to claim 18, wherein this limitation as already been addressed); and

wherein if said parent object is of said fixed object type, defining said new object having a type out of a group comprising said fixed object, said run-time object, said leaf object and said link object (**Refer** to claim 18, wherein this limitation as already been addressed); and

wherein if said parent object is of said run-time object type, defining said new object having a type out of a group comprising said fixed object, said leaf object and said link object (**Refer** to claim 18, wherein this limitation as already been addressed);

defining properties of said new object (**Refer** to claim 18, wherein this limitation as already been addressed);

wherein said plurality of objects serves as template object for deriving corresponding nodes (**Refer** to claim 18, wherein this limitation as already been addressed);

wherein said hierarchical object structure serves as a template structure for deriving hierarchical node structures (**Refer** to claim 18, wherein this limitation as already been addressed); and

wherein said hierarchical node structure comprising a plurality of said nodes is employed for distributing management related information of an electronic device among said plurality of nodes, certain parts of said management related information being assigned to at least one of said plurality of nodes (**Refer** to claim 18, wherein this limitation as already been addressed).

Claim 20 (**New**):

Regarding Claim 20, the combination of Andrews in view of Jőnsson teaches a management system, comprising:

a managed mobile communication enabled device (**SEE** non-patent literature titled – "SyncML-Getting the mobile Internet in sync", **PAGE 112,** wherein subtitle "Optimized for the mobile environment", wherein SyncML has been optimized to meet the specific requirements of the mobile communications environment, taking into account the limited resources of mobile devices and networks and still offering one hundred percent performance on other transport systems such as cable systems or traditional local area networks, Jőnsson); and a hierarchical object structure comprising a plurality of objects being hierarchically associated (Figures 2 and 3, all features, wherein a tree-like hierarchically structures are defined; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network Andrews),

wherein each object of said plurality of objects has a certain object type, which is an object type out of the group comprising at least fixed object, run-time object, leaf object and link object (**REFER** to claim 18, wherein this limitation has already been addressed);

wherein said fixed object type has a fixed title, and wherein said run-time object has a title defined during run-time (**REFER** to claim 18, wherein this limitation has already been addressed);

wherein if a parent object, which is directly arranged superordinate to at least one new object to be defined and is part of said hierarchical object structure has said fixed object type (column 8, lines 29-33; column 5, lines 14-19, wherein a schema may define the classes for a whole directory, or a portion of a directory, wherein the schema includes the classes as well as the containment rules defining the parent/child relationships that are permitted along objects, Andrews), said at least one new object has a type out of a group comprising said fixed object type, said run-time object type, said leaf object type and said link object type (**REFER** to claim 18, wherein this limitation is substantially the same/or similar); and

wherein if said parent object has said run-time object type, said at least one new object has a type out of a group comprising said fixed object type, said leaf object type and said link object type (**REFER** to claim 18, wherein this limitation has already been addressed);

wherein said hierarchical object structure being constituted by said plurality of objects describes and allows generation of a hierarchical node structure constituted by a plurality of nodes (column 6, lines 53-67, wherein organization types objects, diagrams 43 and 44 may represent a logical or physical division of an organization, diagrams 43 and 44 may each represent a different country in which organization maintains an office, subordinate to object, diagram 44 are organization unit type container objects, diagrams 45 and 46, wherein additional objects would typically be subordinate to Organization Unit-Type objects, diagrams 45 and 46, wherein each branch would typically terminate with one or more leaf objects that represent network entities, Andrews);

wherein said hierarchical node structure comprising said plurality of nodes is employed for distributing management related information of said managed mobile communication enabled device among said plurality of nodes, certain parts of said management related information being assigned to at least one of said plurality of nodes (column 4, lines 37-44, wherein hierarchical tree structure simplifies administration entities, wherein a directory service products that uses such a tree like structure for network entity administration is Novell Corporation; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network, wherein frequently distributed meaning portions are stored on different servers; column 10, lines 11, wherein hard disk is equivalent to an electronic device, Andrews);

wherein said managed mobile communication enabled device comprises a device management agent configured to generate at least a part of said hierarchical node structure from said hierarchical object structure to establish said part of said hierarchical node structure and to implement said part of said hierarchical node structure into said managed mobile communication enabled device (**SEE** non-patent literature titled – "SyncML- Getting the mobile Internet in sync", **PAGE 113,** wherein subtitle " Synchronization protocol", wherein synchronization client agent and usually sends the SyncML messages, i.e. operations, possibly including pay load data; **PAGE 113,** wherein subtitle "SyncML framework", wherein the synch server agent manages sync engine access to the neutral and communicates the data synchronization operations to an from the client application, and wherein the sync server agent performs these task by invoking functions in the SyncML I/F, which is the application program interface API to the SyncML adapter, and wherein the APP B uses the client agent to access the network and its SyncML adapter by invoking functions in the SyncML I/F. Jŏnsson);

wherein said device management agent is configured to distribute management related information among said plurality of nodes constituting said hierarchical node structure (column 5, lines 33-48, wherein organization-wide network of local area networks and wide are networks, wherein directories are frequently distributed meaning portions of the directory are stored on different servers, wherein the directory tree illustrates hierarchical tree of various objects associated with the organization network, Andrews); and wherein said device management agent is configured to retrieve at least parts of said management related information from one or more nodes of said plurality of nodes for configuring functions of at least one out of said managed mobile communication enabled device and applications running on said managed mobile communication enabled device to be operable (**SEE** non-patent literature titled – "SyncML- Getting the mobile Internet in sync", **PAGE 110**, subtitle " Synchronization a real business need", wherein with the any device synchronization capabilities of SyncL a mobile user who say receives a order via email can access the company stock inventory system con the same device to determine availability and delivery date ad wherein SyncML enables synchronization to be performed in a standardized way across application, devices, and networks whether it is synchronizing email in a pocket P with that of a stand alone PC via a Blooth Tooth or cable connection or calendar entries on a mobile phone with those of the corporate network via the public mobile phone network, Jőnsson).

Claim 21 (**New**):

Regarding Claim 21, the combination of Andrews in view of Jőnsson teaches a management system, comprising:

a managed mobile communication enabled device (**Refer** to claim 20, wherein this limitation has already been addressed); and

a hierarchical node structure comprising a plurality of nodes for distributing management related information of said managed mobile communication enabled device among said plurality of nodes, certain parts of said management related information being assigned to at least one of said plurality of nodes (**Refer** to claim 20, wherein these limitation are substantially the same/or similar);

wherein each node of said plurality of nodes has a certain node type which is a node type out of the group comprising at least fixed node, run-time node, leaf node and link node (**Refer** to claim 18 wherein these limitation are substantially the same/or similar);

wherein said fixed object type has a fixed title, and wherein said run-time object has a title defined during run-time (**Refer** to claim 18 wherein these limitation are substantially the same/or similar);

wherein if a parent node, which is directly arranged superordinate to one node and is part of said hierarchical node structure, has said fixed node type, said one node has a type out of a group comprising said fixed node type, said run-time node type, said least node type and said link node type (<u>**Refer**</u> to claim 19 wherein these limitation are substantially the same/or similar); and

wherein if said parent node has said run-time node type, said one node has a type out of a group comprising said fixed node type, said leaf node type and said link node type (<u>**Refer**</u> to claim 18 wherein these limitation are substantially the same/or similar);

wherein said managed mobile communication enabled device comprises a device management agent configured to distribute management related information among said plurality of nodes (column 5, lines 33-48, wherein organization-wide network of local area networks and wide are networks, wherein directories are frequently distributed meaning portions of the directory are stored on different servers, wherein the directory tree illustrates hierarchical tree of various objects associated with the organization network, Andrews);

wherein said device management agent is configured to retrieve at least parts of said management related information from one or more nodes of said plurality of nodes (column 4, lines 37-44, wherein hierarchical tree structure simplifies administration entities, wherein a directory service products that uses such a tree like structure for network entity administration is Novell Corporation; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network, wherein frequently distributed meaning portions are stored on different servers; column 10, lines 11, wherein hard disk is equivalent to an electronic device, Andrews); and

wherein said device management agent is for configuring functions of at least one out of said managed mobile communication enabled device and applications running on said managed mobile communication enabled device to be operable (**SEE** non-patent literature titled – "<u>SyncML- Getting the mobile Internet in sync</u>", **PAGE 110**, subtitle " <u>Synchronization a real business need</u>", wherein with the any device synchronization capabilities of SyncL a mobile user who say receives a order via email can access the company stock inventory system con the same device to determine availability and delivery date ad wherein SyncML enables synchronization to be performed in a standardized way across application, devices, and networks whether it is synchronizing email in a pocket P with that of a stand alone PC via a Blooth Tooth or cable connection or calendar entries on a mobile phone with those of the corporate network via the public mobile phone network, Jőnsson).

Claim 22 (**New**):

Regarding Claim 22, teaches an apparatus, comprising:

means for defining at least one new object to be included into a hierarchical object structure comprising a plurality of objects being hierarchically associated;

means for storing data; and

means for interfacing communications (**Refer** to claim 19, wherein this limitation is substantially the same/or similar);

wherein said plurality of objects comprising different types of objects out of a group including at least a fixed object, a run-time object, a leaf object and a link object (**Refer** to claim 18, wherein this limitation is substantially the same/or similar),

wherein said fixed object type has a fixed title, and wherein said run-time object has a title defined during run-time (**Refer** to claim 18, wherein this limitation is substantially the same/or similar);

wherein said apparatus further comprises:

means for checking said type of said parent object (**Refer** to claim 18, wherein this limitation is substantially the same/or similar); and

wherein if said parent object is of said fixed object type, defining said new object having a type out of a group comprising said fixed object, said run-time object, said leaf object and said link object (**Refer** to claim 18, wherein this limitation is substantially the same/or similar); and

wherein if said parent object is of said run-time object type, defining said new object having a type out of a group comprising said fixed object, said leaf object and said link object (**Refer** to claim 18, wherein this limitation is substantially the same/or similar);

means for defining properties of said new object;

wherein said plurality of objects serves as template object for deriving corresponding nodes (Figures 4A and 4B, all features; column 12, lines 45-55, wherein a result is produces by if a source tree and destination tree are the same tree and the move is within the same schema but the move would otherwise prohibit because of containment rule violation, wherein certain steps can be omitted, if the move of the subtree will cross schemas but would not be prohibited because of containment rule violations it is possible certain steps can be omitted, in which the following is interpreted to be deriving, and wherein deriving is equivalent to producing a result, Andrews).;

wherein said hierarchical object structure serves as a template structure for deriving hierarchical node structures (Figures 4A and 4B, all features; column 12, lines 45-55, wherein a result is produces by if a source tree and destination tree are the same tree and the move is within the same schema but the move would otherwise prohibit because of containment rule violation, wherein certain steps can be omitted, if the move of the subtree will cross schemas but would not be prohibited because of containment rule violations it is possible certain steps can be omitted, in which the following is interpreted to be deriving, and wherein deriving is equivalent to producing a result, Andrews).; and

wherein said hierarchical node structure comprising a plurality of said nodes is employed for distributing management related information of an electronic device among said plurality of nodes, certain parts of said management related information being assigned to at least one of said plurality of nodes ((column 4, lines 37-44, wherein hierarchical tree structure simplifies administration entities, wherein a directory service products that uses such a tree like structure for network entity administration is Novell Corporation; column 5, lines 33-53, wherein hierarchical tree of various objects associated with the organization network, wherein frequently distributed meaning portions are stored on different servers; column 10, lines 11, wherein hard disk is equivalent to an electronic device, Andrews);.

Examiner's Response

Applicant argues prior fail to teach, "*fixed object or a run-time object*".

Examiner respectfully disagrees. Referring to column 8, lines 29-33, wherein type object would violate containment rules associated with destination directory, the object type of object is changed to an UNRESTRICTED object, type, wherein UNRESTRICTED object type is a special object type that can exist anywhere in the tree without violating containment rules, wherein the UNRESTRICTED object type is interpreted to be fixed because it doesn't violate rules, wherein the containment rules prohibit certain parent/child relationships among the objects, wherein rules are predefined. In regards to applicant argument "run-time object", Refer to columns 5-6, lines 54-67 and lines 1-3, wherein one or more running processes executing in the run-time environment and directory tree can comprise hundreds or thousands of different objects, wherein process is interpreted to be a running instance of a program.

Applicant argues prior fail to teach, "*an object type that has a fixed title*".

Examiner respectfully disagrees. Applicant argues new claim language, which was not presently defined within the original office action, therefore in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *an object type that has a* ***fixed title***) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant argues prior fail to teach, "*substitution of two objects by a new object*".

Examiner respectfully disagrees. Applicant argues new claim language, which was not presently defined within the original office action, therefore in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *substitution of two objects by* ***new object***) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Prior Art of Record

1. Andrews et al. (US Patent No. 6,105,062) - discloses a method and system for moving a subtree of objects from a source location to a destination location.

2. Jőnsson et al (Non-Patent Literature, Ericsson Review No. 3, 2001) discloses providing an open industry specification for universal data synchronization.

3. SyncML Device Management Tree and Description (Non-Patent Literature, February 15, 2002, version 1.1, 38 pages).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Point of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Helene Rose whose telephone number is (571) 272-0749. The examiner can normally be reached on 8:00am - 4:30pm Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on (571) 272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Helene Rose
Technology Center 2100
November 13, 2006

DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100